



TECHNISCHE
UNIVERSITÄT
DARMSTADT

ULB

Optimales Energie-Management über verteilte, beschränkte Gradientenverfahren

Zimmermann, Jan; Tatarenko, Tatiana; Willert, Volker et al.
(2019)

DOI (TUprints): <https://doi.org/10.25534/tuprints-00014291>
License: only the rights of use according to UrhG
Publication type: Article
Division: 18 Department of Electrical Engineering and Information Technology
Original source: <https://tuprints.ulb.tu-darmstadt.de/14291>

Methoden

Jan Zimmermann*, Tatiana Tatarenko, Volker Willert und Jürgen Adamy

Optimales Energie-Management über verteilte, beschränkte Gradientenverfahren

Optimal energy-management by distributed, constrained gradient descent

<https://doi.org/10.1515/auto-2019-0064>

Empfangen 7. Juni 2019; angenommen 16. August 2019

Zusammenfassung: Dieser Beitrag beschäftigt sich mit verteilten, beschränkten Gradientenverfahren zur Optimierung eines Energie-Management-Problems. Zwei verschiedene Lösungsstrategien werden betrachtet. Zum einen wird ein Entkopplungsansatz analysiert, bei dem über einen Lagrange-Multiplikatoransatz die Beschränkungen in die Zielfunktion aufgenommen werden. Durch ein Gegenbeispiel wird gezeigt, dass dieses Verfahren nicht in jedem Fall auf das globale Optimum des Energie-Management-Problems konvergieren kann. Die zweite Strategie berücksichtigt Nebenbedingungen über einen Straffunktionsansatz und löst das Problem durch die Push-Sum-Konsensus-Dynamik. In der anschließenden Analyse dieses Verfahrens durch Simulation wird auf die Problematik der optimalen Parameterwahl sowie auf das Konvergenzverhalten bei unterschiedlicher Knoten- und Kantenanzahl des Graphen eingegangen.

Schlagwörter: Verteilte Optimierung, Konsensus, Smart Grids

Abstract: This paper deals with distributed, constrained gradient descents in application to the optimization of an energy-management-problem. Two different solution strategies are considered. First, a decoupling approach is analyzed that employs a Lagrange approach to include the constraints in the objective function. By means of a counterexample it is shown that this procedure does not lead to the global optimum of the considered energy-management-problem in every case. The second strategy

incorporates constraints by means of penalty-functions and solves the problem using the push-sum-consensus. The ensuing analysis by simulation is concerned with the difficulty of identifying the optimal parameter set and examines the convergence behavior with regard to different node and edge numbers of distinct communication graphs.

Keywords: Distributed optimization, consensus, smart grids

1 Einleitung

Smart Grids sind die Zukunft von heutigen elektrischen Netzwerken. Durch die Weiterentwicklung soll es unter anderem möglich werden, das Verhalten aller Akteure im Netzwerk durch genaue Vorhersagen besser zu koordinieren. Dazu wird die bestehende Infrastruktur durch moderne Sensorik, Kontrollmethoden und integrierte Kommunikation erweitert, wodurch die Stromversorgung effizienter, verlässlicher, ökonomischer und nachhaltiger wird [1].

Ein Trend, der sich derzeit abzeichnet, ist die Entwicklung weg von zentralen Organisationseinheiten hin zu dezentralen und verteilten Instanzen, die jeweils Kontrolle über eine bestimmte Teilaufgabe oder ein bestimmtes lokales Gebiet übernehmen. Die Aktionen dieser Instanzen sind untereinander verkoppelt, beispielsweise durch das elektrische Übertragungsnetz. Aufgrund dieser Verzahnung ist Zusammenarbeit von Akteuren und die Kommunikation lokaler Information unerlässlich [6]. Um optimale Aktionen und Strategien in diesen Netzwerken zu identifizieren, ist es erforderlich, Optimierungsprobleme verteilt und ohne zentrale Kontrolleinheit lösen zu können. Neben der natürlichen Eingliederung in entstehende Strukturen bieten verteilte Methoden die Vorteile von Skalierbarkeit, Robustheit und Kommunikationseffizienz [18]. Die grundlegenden Ziele der elektrischen Energieversorgung bleiben auch bei diesem Strukturwandel gleich: Es gilt Produktion, Übertragung und Verbrauch zu optimieren [1]. Dazu wird unter anderem das Energie-Management

*Korrespondenzautor: Jan Zimmermann, Technische Universität Darmstadt, Fachbereich Elektrotechnik und Informationstechnik, Fachgebiet Regelungsmethoden und Robotik, D-64283 Darmstadt, Deutschland, E-Mail: jan.zimmermann@rmr.tu-darmstadt.de
Tatiana Tatarenko, Volker Willert, Jürgen Adamy, Technische Universität Darmstadt, Fachbereich Elektrotechnik und Informationstechnik, Fachgebiet Regelungsmethoden und Robotik, D-64283 Darmstadt, Deutschland, E-Mails: tatiana.tatarenko@rmr.tu-darmstadt.de, vwillert@rmr.tu-darmstadt.de, adamy@rmr.tu-darmstadt.de

als Optimierungsproblem definiert, bei dem eine kosteneffiziente Befriedigung der Elektrizitätsnachfrage im Vordergrund steht.

In dieser Arbeit wird ein solches Energie-Management-Problem für ein verteiltes Szenario betrachtet, bei dem die von mehreren Generatoren produzierte Leistung auf die von Verbrauchern nachgefragte Leistung angepasst wird. Dabei werden zusätzlich Verluste berücksichtigt, die kraftwerksspezifisch sind und bei jedem realen Generator auftreten. Neben der monetären Effizienz der Generatoren wird die Befriedigung der Verbrauchernachfrage optimiert, die daher nicht als fest vorgegeben angenommen wird. Damit entspricht das Problem einer sogenannten *Wohlfahrtsoptimierung*, bei der die einzelnen Akteure mit einander kooperieren, um die Kosten bzw. Nutzen zu minimieren bzw. zu maximieren. Das Problem beinhaltet somit einen *Demand-Response-Ansatz*, bei denen Verbraucher durch den Elektrizitätspreis so beeinflusst werden, dass sie von ihrem normalen und global betrachtet ineffizienten Verbrauchsmuster abweichen. Durch diesen zusätzlichen Freiheitsgrad wird das gesamte System flexibler und volatile Stromproduzenten, wie zum Beispiel Windkraft- und Solaranlagen, können besser in das bestehende Netz integriert werden [13].

Für die Formulierung des verteilten Energie-Management-Problems eignet sich ein Multi-Agenten-System Ansatz [18]. Hierbei stellen einzelne Generatoren und Verbraucher die Agenten im Netzwerk dar, die Informationen untereinander austauschen können. Wie in Multi-Agenten-Systemen üblich, werden die Kommunikationsverbindungen durch einen Graphen beschrieben, wobei die Knoten den Agenten und die Kanten den jeweiligen Kanälen entsprechen. Das betrachtete Optimierungsproblem wird als verteilt bezeichnet, da jeder Agent nur über lokale Informationen verfügt, auf die keiner der restlichen Agenten direkten Zugriff hat. Um das Problem, welches in Abschnitt 3 genauer definiert wird, zu lösen, ist ein effizienter Informationsaustausch zwischen den Agenten erforderlich. Hierfür wird bei den in dieser Veröffentlichung behandelten Algorithmen eine Konsensus-Dynamik verwendet.

In [8] und [12] werden verteilte Gradientenverfahren für unbeschränkte Optimierungen betrachtet. [8] verwendet das Push-Sum-Konsensus-Protokoll zur Berechnung von verteilten Informationen und beweist, dass eine Konvergenz auf ein Optimum auch bei zeitlichen Änderungen der Kommunikationsverbindungen möglich ist. [12] untersucht ein Push-Pull-Konsensus-Protokoll zur Lösung eines verteilten Optimierungsproblems und kann eine lineare Konvergenz des dort vorgestellten Algorithmus nachweisen. [5] untersucht ebenfalls unbeschränkte Gradi-

entenverfahren, die auf einer verteilten Ausführung des Nesterov-Gradienten-Verfahrens beruhen und dadurch eine schnelle Konvergenzrate aufweisen.

In [9] werden Konsensus- und verteilte Gradientenmethoden so erweitert, dass Nebenbedingungen bei der Optimierung berücksichtigt werden können. Hierfür wird eine Projektionsmethode verwendet, bei der ein Aktualisierungsschritt des Gradientenabstiegs auf die erlaubte Lösungsmenge projiziert wird. Voraussetzung für die Anwendbarkeit des Projektionsverfahrens ist hier ein ungerichteter Kommunikationsgraph. Die Veröffentlichung [15] verwendet ebenfalls eine Projektionsmethode. Als Kommunikationssteuerung wird das Push-Sum-Protokoll eingesetzt, wodurch eine Konvergenz des Verfahrens auch auf gerichteten Graphen möglich wird. Um die Projektion der Optimierungsvariablen auf die erlaubte Lösungsmenge durchführen zu können, wird hier allerdings gefordert, dass alle Nebenbedingungen global bekannt und daher nur die Kostenfunktionen lokal sind. Dies stellt eine erhebliche Einschränkung der Menge der mit dieser Methode lösbaren Optimierungsprobleme dar.

[14] stellt schließlich einen Algorithmus vor, der auf gerichteten und zeitvarianten Kommunikationsgraphen zur optimalen Lösung konvergiert und theoretisch alle konvexen Nebenbedingungen berücksichtigen kann, unabhängig davon, ob sie lokal oder global definiert sind. Dazu wurde das Push-Sum-Gradientenverfahren aus [8] durch das Berücksichtigen von Nebenbedingungen durch Straffunktionen in der Zielfunktion erweitert. Es wird darüber hinaus bewiesen, dass bei einem bestimmten Verhältnis von Gradientenschrittweite zu Straffunktionsparameter das Verfahren zuverlässig auf einen optimalen Wert konvergiert. Der erste Beitrag dieses Artikels stellt die Analyse des in [14] beschriebenen Algorithmus in Anwendung auf ein Energie-Management-Problem dar. Eingegangen wird hierbei auf eine angemessene Wahl der offenen Parameter des Verfahrens und auf die Konvergenzgeschwindigkeit in Abhängigkeit von Kommunikationsgraphen mit unterschiedlichen Vernetzungsgraden und Agentenanzahlen.

Eine weitere Möglichkeit, beschränkte und verteilte Optimierungsprobleme zu lösen, wird in [18] vorgestellt. Hier wird explizit zwischen lokalen und globalen Anteilen des Optimierungsproblems unterschieden. Der lokale Anteil kann durch eine zentrale Optimierungsmethode am jeweiligen Knoten gelöst werden, bei welcher der globale Anteil über eine Konsensus-Dynamik berücksichtigt wird. Damit kann ein wesentlicher Teil des Optimierungsproblems lokal gelöst werden, wodurch sich die Anzahl der notwendigen Kommunikationsschritte gering halten

lässt. Insgesamt handelt es sich um einen vielversprechenden Ansatz. In der vorliegenden Veröffentlichung wird jedoch anhand eines Gegenspiels gezeigt, dass der in [18] vorgestellte Algorithmus nicht in jedem Fall auf den optimalen Wert konvergiert und daher andere Methoden, wie zum Beispiel der in [14] veröffentlichte Algorithmus, zur Lösung des Problems herangezogen werden müssen. Dies stellt den zweiten Beitrag dar.

Die Veröffentlichung gliedert sich wie folgt. In Abschnitt 2 werden kurz die für diese Arbeit relevanten Notationen und Graphbeschreibungen vorgestellt. Abschnitt 3 stellt das untersuchte Energie-Management-Problem vor und zeigt, welche Transformationen durchgeführt werden müssen, damit das Optimierungsproblem von dem in Abschnitt 5 vorgestellten Algorithmus gelöst werden kann. Der Abschnitt 4 geht auf einen weiteren, bereits veröffentlichten Lösungsansatz ein und zeigt durch ein Gegenbeispiel, dass dieser Algorithmus nicht immer auf den optimalen Wert konvergiert. In Abschnitt 6 wird die Lösung des beschriebenen Energie-Management-Problems durch den verteilten Straffunktionsansatz mittels Simulationen untersucht. Abschließend werden die Ergebnisse der Veröffentlichung in Abschnitt 7 zusammengefasst.

2 Notation und Graphen

Folgende Notationen werden verwendet: Die Menge $\{1, \dots, n\}$ wird durch $[n]$ beschrieben. Vektoren \mathbf{v} in multidimensionalen Räumen werden durch Fettdruck von Skalaren s unterschieden. Alle Zeitindizes, wie zum Beispiel t , gehören zur Menge nicht-negativer ganzer Zahlen \mathbb{Z}^+ . Die Funktion $1_{\mathcal{A}}(x)$ ist die Indikatorfunktion der Menge \mathcal{A} , wobei $1_{\mathcal{A}}(x) = 1$, wenn $x \in \mathcal{A}$ und $1_{\mathcal{A}}(x) = 0$, sonst gilt. $|\mathcal{N}|$ bezeichnet die Kardinalität der Menge \mathcal{N} .

In dieser Arbeit wird ein gerichteter Graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ verwendet, um die Kommunikationsverbindungen zwischen den Agenten zu beschreiben. Die Menge der Agenten ist hierbei mit der Menge der Knoten im Graphen \mathcal{G} gleichgesetzt und wird durch \mathcal{V} beschrieben. Die Menge $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ entspricht den Kanten des Graphen. Es existiert eine Kommunikationsverbindung von Knoten j zu Knoten i , d. h., j kann Informationen an i senden, wenn $(j, i) \in \mathcal{E}$. Die Eingangsnachbarschaft von Knoten i wird mit \mathcal{N}_i^- bezeichnet und beinhaltet alle Knoten, die Informationen an den Knoten i senden können. Der Eingangsgrad von i entspricht der Kardinalität der Eingangsnachbarschaft $|\mathcal{N}_i^-|$. Entsprechend bezeichnet \mathcal{N}_i^+ die Ausgangsnachbarschaft von i und $|\mathcal{N}_i^+|$ den Ausgangsgrad. In dieser Arbeit werden ausschließlich stark verbundene Graphen

berücksichtigt, d. h., dass zwischen jedem Knoten des Graphen \mathcal{G} ein gerichteter Pfad zu jedem anderen Knoten existiert. Diese Eigenschaft ist unerlässlich, wenn alle Knoten als gleichwertige Agenten im Netzwerk aktiv sein sollen, da bei nicht stark verbundenen Graphen immer eine Untermenge an Knoten existiert, die entweder keine Informationen in das restliche Netzwerk einspeisen oder keine Informationen aus dem restlichen Netzwerk empfangen kann.

3 Energie-Management-Problem

Das hier zu behandelnde Energie-Management-Problem wurde in der Veröffentlichung [18] vorgestellt und in [14] weiter analysiert. In diesem speziellen Problem wird die Einsatzplanung von Erzeugern und Verbrauchern betrachtet, wobei die Minimierung der Produktionskosten bei angemessener Befriedigung der Nachfrage die Festlegung der Optimierungsvariable bestimmt.

Damit werden zwei unterschiedliche Akteure betrachtet: Generatoren $i \in \mathcal{N}_G$ und Verbraucher $j \in \mathcal{N}_V$. Die Aufgabe der Generatoren ist es, die von den Verbrauchern nachgefragte Leistung zu produzieren und zusätzlich Leistungsverluste zu berücksichtigen. Der Erlös $K(p)$ bei Verkauf der Leistungsmenge p lässt sich laut [18] mit einer quadratischen Funktion approximieren. Beschränkt wird die produzierbare Leistungsmenge durch eine obere und untere Schranke

$$\underline{p}_i \leq p_i \leq \overline{p}_i \text{ für alle } i \in \mathcal{N}_G. \quad (1)$$

In [14] wird zudem eine Erweiterung der Kostenfunktion vorgeschlagen, die garantiert, dass der Gradient der Funktion außerhalb des erlaubten Bereiches $[\underline{p}_i, \overline{p}_i]$ einen konstanten Wert annimmt und dadurch begrenzt bleibt. Das entspricht einer Standardannahme in verteilten Gradientenverfahren [4] [5] [8] [9]. Es ergibt sich die Kostenfunktion

$$K_i(p_i) = \begin{cases} a_i p_i^2 + b_i p_i + c_i, & p_i \in [\underline{p}_i, \overline{p}_i], \\ (2a_i \underline{p}_i + b_i) p_i, & p_i < \underline{p}_i, \\ (2a_i \overline{p}_i + b_i) p_i, & p_i > \overline{p}_i, \end{cases} \quad (2)$$

wobei a_i , b_i und c_i generatorspezifische Parameter sind.

Die Zielfunktion $U(p)$ der Verbraucher richtet sich nach der Befriedigung der jeweiligen Leistungsnachfrage \overline{P} . Dabei wird davon ausgegangen, dass sich diese maximale Nachfrage theoretisch bis zu einem minimalen Wert

von \underline{p} absenken lässt, wobei allerdings der Ausgabewert von $\bar{U}(\underline{p})$ sinkt. Damit gilt

$$\underline{p}_j \leq p_j \leq \bar{P}_j \text{ für alle } j \in \mathcal{N}_V. \quad (3)$$

Hintergrund dieser Dynamik ist die mögliche Absenkung der eigenen Nachfrage zur Reduktion der Anforderungen an die Generatoren und damit eine Kostenreduktion für das gesamte Netzwerk. In [14] wird die gegenüber [18] leicht modifizierte Funktion

$$U_j(p_j) = \begin{cases} w_j p_j - \alpha_j p_j^2, & p_j \leq \frac{w_j}{2K_j \alpha_j}, \\ (w_j - \frac{w_j}{K_j}) p_j - \frac{w_j^2}{4K_j^2 \alpha_j}, & p_j > \frac{w_j}{2K_j \alpha_j}, \end{cases} \quad (4)$$

vorgeschlagen. Auch hier sind die Parameter w_j , α_j und $0 < K_j < 1$ spezifisch für jeden Verbraucher.

Bei Betrachtung der Zielfunktionen für Generatoren und Verbraucher fällt auf, dass die einzelnen lokalen Kostenfunktionen ausschließlich von der lokalen Variable p_i , $i \in \mathcal{N}_G$ bzw. p_j , $j \in \mathcal{N}_V$ abhängig sind und sich daher nicht gegenseitig beeinflussen. Die Kopplung der lokalen Optimierungsprobleme entsteht durch die Gleichgewichtsbedingung

$$\sum_{i \in \mathcal{N}_G} (p_i - l_i p_i^2) = \sum_{j \in \mathcal{N}_V} p_j, \quad (5)$$

die sicherstellt, dass die produzierte Leistungsmenge der nachgefragten entspricht. Die Leistungsverluste $l_i p_i^2$ der jeweiligen Generatoren hängen von einem spezifischen, konstanten Parameter $0 \leq l_i < \alpha_i$ und der Menge der produzierten Leistung ab.

Ziel dieses Energie-Management-Problems ist es, den Erlös der Generatoren und gleichzeitig die Befriedigung der Nachfrage zu maximieren, was einer Minimierung des folgenden Optimierungsproblems entspricht

$$\min_{\mathbf{p}} \sum_{i \in \mathcal{N}_G} K_i(p_i) - \sum_{j \in \mathcal{N}_V} U_j(p_j), \quad (6)$$

$$\text{s. t. } \sum_{i \in \mathcal{N}_G} (p_i - l_i p_i^2) = \sum_{j \in \mathcal{N}_V} p_j, \quad (6a)$$

$$\underline{p}_i \leq p_i \leq \bar{P}_i, i \in \mathcal{N}_G, \quad (6b)$$

$$\underline{p}_j \leq p_j \leq \bar{P}_j, j \in \mathcal{N}_V. \quad (6c)$$

Dabei wird die kompakte Vektorschreibweise $\mathbf{p} = [p_1, \dots, p_n]$ für die Optimierungsvariablen verwendet. Dieses Problem besitzt somit eine konvexe Zielfunktion und die ebenfalls konvexen Nebenbedingungen (6b) und (6c). Die verkoppelnde Gleichgewichtsbedingung aus Gleichung (6a) ist allerdings nicht konvex.

Um zu garantieren, dass ein Gradientenabstiegsverfahren immer auf das Optimum konvergiert, wird in [14] eine konvexe Reformulierung vorgeschlagen, die neben der Konvexität des Problems auch eine Beschränkung aller Gradienten sicherstellt. Hierfür wird in einem ersten Schritt die Hilfsvariable $v_i = l_i p_i^2$ eingeführt und im zweiten Schritt die Menge der Nebenbedingungen um die konvexe Ungleichung $v_i \geq l_i p_i^2$ erweitert. Es ergibt sich das vollständig konvexe Optimierungsproblem

$$\min_{\mathbf{p}} \sum_{i \in \mathcal{N}_G} K_i(p_i) - \sum_{j \in \mathcal{N}_V} U_j(p_j), \quad (7)$$

$$\text{s. t. } \sum_{i \in \mathcal{N}_G} (p_i - v_i) = \sum_{j \in \mathcal{N}_V} p_j, \quad (7a)$$

$$\underline{p}_i \leq p_i \leq \bar{P}_i, i \in \mathcal{N}_G, \quad (7b)$$

$$\underline{p}_j \leq p_j \leq \bar{P}_j, j \in \mathcal{N}_V, \quad (7c)$$

$$v_i \geq l_i p_i^2, i \in \mathcal{N}_G. \quad (7d)$$

In Proposition 2 aus [14] wird bewiesen, dass die Lösungen der Probleme (6) und (7) äquivalent zueinander sind, wenn folgende Annahme erfüllt ist:

Annahme 1. Die oberen und unteren Grenzen für Leistungsgenerierung und -verbrauch aus Gleichungen (1) und (3) erfüllen die Bedingung

$$\sum_{j \in \mathcal{N}_V} \bar{P}_j \geq \sum_{i \in \mathcal{N}_G} (\bar{p}_i - l_i (\bar{p}_i)^2). \quad (8)$$

Aufgrund der räumlichen Separation der einzelnen Kostenfunktionen kann das beschriebene Problem als verteilte Optimierung interpretiert werden. Hierbei stellen die Generatoren $i \in \mathcal{N}_G$ und die Verbraucher $j \in \mathcal{N}_V$ eigenständige Instanzen mit privaten Kostenfunktionen K_i bzw. U_j dar, deren Ziel es ist, gemeinsam ihre aufsummierten Kosten unter Berücksichtigung genannter Nebenbedingungen zu minimieren. Die konvexen Nebenbedingungen sind dabei folgendermaßen auf die Agenten verteilt:

$$C_i = \begin{cases} c_{i,1}(p_i) & = p_i - \bar{P}_i \leq 0, \\ c_{i,2}(p_i) & = \underline{p}_i - p_i \leq 0, \\ c_{i,3}(\mathbf{p}, \mathbf{v}) & = \sum_{i \in \mathcal{N}_G} (p_i - v_i) - \sum_{j \in \mathcal{N}_V} p_j \leq 0, \\ c_{i,4}(\mathbf{p}, \mathbf{v}) & = -\sum_{i \in \mathcal{N}_G} (p_i - v_i) + \sum_{j \in \mathcal{N}_V} p_j \leq 0, \\ c_{i,5}(p_i, v_i) & = (l_i p_i^2 - v_i) 1_{\{p_i \in [\underline{p}_i, \bar{P}_i]\}} \\ & \quad + (2l_i \underline{p}_i - v_i) 1_{\{p_i < \underline{p}_i\}} \\ & \quad + (2l_i \bar{P}_i - v_i) 1_{\{p_i > \bar{P}_i\}} \leq 0, \end{cases} \quad \text{für alle } i \in \mathcal{N}_G, \quad (9)$$

$$C_j = \begin{cases} c_{j,1}(p_j) & = p_j - \bar{p}_j \leq 0, \\ c_{j,2}(p_j) & = \underline{p}_j - p_j \leq 0, \end{cases} \quad \text{für alle } j \in \mathcal{N}_V. \quad (10)$$

4 Lösungsansatz 1: Entkopplung von lokalen und globalen Subproblemen

In [18] wurde ein Algorithmus veröffentlicht, der explizit die Aufteilung des Problems (7) in einen lokalen und globalen Anteil ausnutzt, um einen Teil des Optimierungsproblems lokal lösen zu können. Dadurch lässt sich theoretisch die Anzahl der Kommunikationsschritte senken. Dazu wird die globale Nebenbedingung (7a) mit dem Lagrange-Faktor λ der Zielfunktion hinzugefügt

$$\mathcal{L}(P, \lambda) = \sum_{i \in \mathcal{N}_G} K_i(p_i) - \sum_{j \in \mathcal{N}_V} U_j(p_j) + \lambda \left(\sum_{j \in \mathcal{N}_V} p_j - \sum_{i \in \mathcal{N}_G} (p_i - l_i p_i^2) \right). \quad (11)$$

Es werden anschließend die lokalen Lagrange-Faktoren λ_i definiert, die durch eine Konsensus-Dynamik auf einen gemeinsamen Wert geführt werden sollen. Nach [18] und dem dort beschriebenen Algorithmus entspricht dieser Konsensuswert dem optimalen Lagrange-Faktor der Lagrangefunktion, d. h., $\lambda_i = \lambda^*$ für alle $i \in \mathcal{V}$. Unter Verwendung der lokalen Lagrange-Faktoren lässt sich die Optimierung in die Lösung der Subprobleme

$$p_i(t+1) = \operatorname{argmin}_{\underline{p}_i \leq p_i(t) \leq \bar{p}_i} [K_i(p_i(t)) - \lambda_i(t+1)p_i(t)], \quad (12)$$

$$p_j(t+1) = \operatorname{argmin}_{\underline{p}_j \leq p_j(t) \leq \bar{p}_j} [\lambda_j(t+1)p_j(t) - U_j(p_j(t))], \quad (13)$$

mit $i \in \mathcal{N}_G$ und $j \in \mathcal{N}_V$ aufspalten. Die Aktualisierung von λ_i erfolgt nach der Regel

$$\lambda_i(t+1) = \sum_{j \in \mathcal{V}} w_{ij} \lambda_j(t) + \eta \xi_i(t), \quad \text{für alle } i \in \mathcal{V}. \quad (14)$$

Hierbei stellt η einen festen Schrittweitenparameter dar und $\xi_i(t)$ beschreibt die am Knoten i festgestellte Leistungsabweichung zwischen Produktion, Verlust und Verbrauch. Für die Aktualisierung dieser Variable wird eine Konsensusdynamik mit einer spaltenstochastischen Matrix verwendet, für die exakte Formulierung sei auf [18] verwiesen. Die Gewichte w_{ij} sind Elemente der zeilenstochastischen Matrix W , die sich aus den Kommunikations-

verbindungen zwischen den einzelnen Agenten ergibt:

$$w_{ij} = \begin{cases} \frac{1}{|\mathcal{N}_i^-|+1}, & j \in \mathcal{N}_i^- \\ 1 - \sum_{j \in \mathcal{N}_i^-} w_{ij}, & j = i \\ 0, & j \notin \mathcal{N}_i^-. \end{cases} \quad (15)$$

Werden die Aktualisierungen für $\lambda_i(t+1)$, $p_i(t+1)$ und $\xi_i(t+1)$ iterativ wiederholt, gilt laut [18]

$$\lim_{t \rightarrow \infty} \lambda_i(t) = \lambda_c, \quad \text{für alle } i \in \mathcal{V} \quad (16)$$

$$\lim_{t \rightarrow \infty} \xi_i(t) = 0, \quad \text{für alle } i \in \mathcal{V} \quad (17)$$

$$\lim_{t \rightarrow \infty} p_i(t) = p_i^*, \quad \text{für alle } i \in \mathcal{V}. \quad (18)$$

λ_c repräsentiert den Konsensuswert der lokalen Lagrange-Multiplikatoren und p_i^* beschreibt den optimalen Wert der Variable p_i .

Dieser Ansatz wurde in [18] auf das in Abschnitt 3 beschriebene Problem angewendet. Durch die Entkopplung findet eine Verschiebung eines Großteils der Berechnungen des Optimierungsproblems in die lokalen Agenten statt. Dadurch erhöhen sich zwar die lokalen Rechenschritte, die Anzahl der notwendigen Kommunikationsschritte bis zur Konvergenz des Algorithmus sinkt jedoch. Damit wurde in [18] gezeigt, wie sich durch geschicktes Ausnutzen der Struktur des Optimierungsproblems eine Verschiebung des Schwerpunkts der Berechnungen erreichen lässt. Ein Nachteil des Verfahrens stellt die zweifach Konsensusbildung über die Variablen λ_i und ξ_i dar. Dadurch sind pro Iteration des Algorithmus zwei Kommunikationsschritte notwendig, während in dem Verfahren aus [14] beispielsweise nur ein Konsensus Schritt berechnet werden muss.

Diese Strukturausnutzung kann in zukünftigen Arbeiten zu interessanten Ergebnissen führen, im Folgenden wird allerdings durch ein Gegenbeispiel gezeigt, dass der Algorithmus aus [18] unter den dort angegebenen Konvergenzbedingungen nicht immer auf den optimalen Wert des Optimierungsproblems (6) bzw. (7) konvergiert.

Gegenbeispiel

In diesem Abschnitt soll ein explizites Szenario betrachtet werden, dass alle Annahmen aus [18] erfüllt, für das der Algorithmus aus [18] allerdings nicht auf den optimalen Wert des zentralen Problems konvergiert.

Betrachtet wird das konvexe Problem aus (7) mit der Kostenfunktion (2) für Generatoren und (4) für Verbraucher. Dieses wird zunächst zentral mit einem Standard-Solver gelöst, um die optimale Lösung als Referenz zu erhalten. Diese Lösung kann nun verwendet werden, um zu

zeigen, dass der im vorherigen Abschnitt beschriebene Algorithmus in dem betrachteten Szenario nicht auf den optimalen Wert konvergieren kann. Um das Gegenbeispiel übersichtlich zu halten, werden ausschließlich die optimalen Werte für die Generatoren überprüft.

Das betrachtete Szenario besteht aus zwei Generatoren und zwei Verbrauchern. Die Parametrierung wird aus [18] entnommen und ist in Tabelle 1 in Abschnitt 6 aufgeführt. Verbraucher und Generatoren haben hierbei jeweils die IDs 1 und 2. Für die Generatoren ergeben sich die optimalen, zentral berechneten Werte

$$\mathbf{p}^{*,\text{zentral}} = \begin{bmatrix} 81,9772 \\ 124,8043 \end{bmatrix}. \quad (19)$$

Für das Gegenbeispiel wird der auskonvergierte Fall, d. h., $t \rightarrow \infty$, des Algorithmus betrachtet. Damit gelten die Grenzwerte aus den Gleichungen (16) bis (18). Explizit sei an dieser Stelle darauf hingewiesen, dass für diesen Fall auf Grund der Konsensus-Dynamik alle λ_i den gemeinsamen, konstanten Wert λ_c annehmen.

Die Änderungen der Optimierungsvariablen der Generatoren erfolgen ausschließlich über die Lösung des Subproblems aus Gleichung (12). Die letzte Aktualisierung vor Beenden des Algorithmus erfolgt unter Verwendung von λ_c . Aufgrund der Tatsache, dass der optimale Wert des zentral gelösten Problems zwischen den oberen und unteren Grenzen liegt,

$$\underline{p}_i < p_i^{*,\text{zentral}} < \overline{p}_i, \quad (20)$$

müsste der optimale Wert des Algorithmus das unbeschränkte Optimierungsproblem

$$p_i^{*,[18]} = \operatorname{argmin}_{p_i} a_i p_i^2 + b_i z_i + c_i - \lambda_c z_i, \text{ für alle } i \in \mathcal{N}_G \quad (21)$$

lösen. An dieser Stelle wird nun die Annahme gemacht, dass die Lösung des Algorithmus aus [18] dem tatsächlichen Optimum des Problems entspricht, d. h.

$$\mathbf{p}^{*,\text{zentral}} = \mathbf{p}^{*,[18]}. \quad (22)$$

Nun wird durch Einsetzen der Lösung $p_i^{*,\text{zentral}}$ in die Gleichungen des Algorithmus der Konsensuswert λ_c berechnet und anschließend gezeigt, dass die Annahme nicht korrekt ist.

Aufgrund der strikten Konvexität dieses Problems wird das einzige Minimum der Zielfunktion erreicht, wenn für die erste Ableitung

$$2a_i p_i^{*,\text{zentral}} + b_i - \lambda_c = 0 \quad (23)$$

gilt. Wird diese Gleichung nach λ_c aufgelöst und der Wert der zentralen Lösung des Problems eingesetzt, ergibt sich unter Verwendung der entsprechenden Parametrierung

$$\begin{aligned} \lambda_c &= 2 \cdot 0,0024 \cdot 81,9772 + 5,56 \\ &= 5,9535 \text{ für } i = 1 \end{aligned}$$

und

$$\begin{aligned} \lambda_c &= 2 \cdot 0,0056 \cdot 124,8043 + 4,32 \\ &= 5,7178 \text{ für } i = 2. \end{aligned}$$

Damit gilt

$$\lambda_c \neq \lambda_c,$$

was direkt der Konsensus-Dynamik aus Gleichung (16) und damit der Aussage von Theorem 3 aus [18] widerspricht.

Damit ist gezeigt, dass in diesem Szenario

$$\mathbf{p}^{*,\text{zentral}} \neq \mathbf{p}^{*,[18]} \quad (24)$$

gilt und somit der Algorithmus nicht auf den optimalen Wert konvergieren kann.

Es bleibt zu untersuchen, ob das Verfahren so angepasst werden kann, dass für alle Fälle das Optimierungsproblem korrekt gelöst wird, da der Entkopplungsansatz und damit die Verschiebung der Berechnungen in die einzelnen Agenten durchaus ein interessanter Ansatz ist.

5 Lösungsansatz 2: Verteilter Gradientenabstieg mit Straffunktionen

Unter Verwendung der Push-Sum-Konsensus-Dynamik wurde in [8] ein Verfahren vorgestellt, das ein allgemeines, unbeschränktes und konvexes Optimierungsproblem löst. In [14] wurde dieses Verfahren aufgegriffen und erweitert, sodass nun zusätzlich Nebenbedingungen für dieses Problem berücksichtigt werden können und somit das verteilte Optimierungsproblem (7) gelöst werden kann. Um die Funktionsweise des Verfahrens zu erklären, wird zunächst das allgemeine, verteilte Optimierungsproblem

$$\min_{\mathbf{z}} F(\mathbf{z}) = \sum_{i=1}^n F_i(\mathbf{z}), \mathbf{z} \in \mathbb{R}^d, \quad (25)$$

$$\mathbf{s.t.} \ c_1(\mathbf{z}) \leq 0, c_2(\mathbf{z}) \leq 0, \dots, c_n(\mathbf{z}) \leq 0, \quad (25a)$$

mit den konvexen Funktionen $F_i : \mathbb{R}^d \rightarrow \mathbb{R}$ und konvexen Nebenbedingungen $c_i : \mathbb{R}^d \rightarrow \mathbb{R}$, $i \in [n]$ betrachtet. Dieses

durch die lokalen Ungleichungen $c_i(\mathbf{z}) \leq 0$ beschränkte Problem wird zunächst unter Verwendung eines Straffunktionsansatzes in ein unbeschränktes transformiert. Dazu werden die Nebenbedingungen über konvexe Straffunktionen $\Psi_i(\mathbf{z})$ der Zielfunktion über die Parameterfolge r_t hinzugefügt, deren Werte sich mit voranschreitender Iterationsanzahl erhöhen und somit mehr und mehr Gewicht auf die Erfüllung der Nebenbedingung gelegt wird. Unter Verwendung von

$$\Psi_i(\mathbf{z}) = g(c_i(\mathbf{z})), \quad (26)$$

$$\text{mit } g(u) = \begin{cases} \log\left(\frac{e^u + e^{-u}}{2}\right), & u > 0, \\ 0, & u \leq 0, \end{cases} \quad (27)$$

lässt sich ein entsprechendes verteiltes und konvexes Optimierungsproblem durch

$$\min_{\mathbf{z}} F_t(\mathbf{z}) = \min_{\mathbf{z}} \sum_{i=1}^n [F_i(\mathbf{z}) + r_t \Psi_i(\mathbf{z})] \quad (28)$$

ausdrücken. Die Gesamtkostenfunktion des Optimierungsproblems $F_t(\mathbf{z})$ ändert sich entsprechend der Folge r_t mit der Zeit. Damit liegt nun ein unbeschränktes Optimierungsproblem vor, bei dem eine Nichterfüllung einer Nebenbedingung einen Anstieg der Zielfunktion zur Folge hat.

Ein praktischer Nachteil des Straffunktionsverfahren ist, dass das Optimum \mathbf{z}^* erst dann exakt erreicht wird, wenn der Straffunktionsparameter r_t unendlich groß wird [11], d. h.,

$$\mathbf{z}(t) \rightarrow \mathbf{z}^*, \text{ wenn } r_t \rightarrow \infty, \quad (29)$$

wobei der Vektor $\mathbf{z}(t)$ die Optimierungsvariablen zum Zeitpunkt t beschreibt. Daher wird bei Verwendung von Straffunktionsbeschränkungen ein Optimum in der Regel nur annähernd erreicht und es ist eine hohe Iterationsanzahl notwendig, um eine geringe Abweichung zum Optimum zu realisieren. Auf diese Tatsache wird auch bei der praktischen Analyse des Verfahrens in Abschnitt 6 eingegangen.

Die Basis des Algorithmus, der in [14] vorgeschlagen wird, bildet das Push-Sum- oder Ratio-Protokoll [2] [7], mit dem effizient Mittelwerte aus verteilten Startzuständen berechnet werden können. Dabei werden die Vektorvariablen $\mathbf{z}_i(t)$, $\mathbf{x}_i(t)$, $\mathbf{w}_i(t) \in \mathbb{R}^d$ und die skalare Variable $y_i(t)$ iterativ aktualisiert, wobei der Anfangswert letzterer zu $y_i(0) = 1$, $\forall i \in [n]$ gewählt werden muss. Nach folgender Vorschrift wird der Gradient in dieses Kommunikationsgerüst eingegliedert:

$$\mathbf{w}_i(t+1) = \frac{1}{|\mathcal{N}_i^+| + 1} \mathbf{x}_i(t) + \sum_{j \in \mathcal{N}_i^-} \frac{1}{|\mathcal{N}_j^+| + 1} \mathbf{x}_j(t), \quad (30)$$

$$y_i(t+1) = \frac{1}{|\mathcal{N}_i^+| + 1} y_i(t) + \sum_{j \in \mathcal{N}_i^-} \frac{1}{|\mathcal{N}_j^+| + 1} y_j(t), \quad (31)$$

$$\mathbf{z}_i(t+1) = \frac{\mathbf{w}_i(t+1)}{y_i(t+1)}, \quad (32)$$

$$\mathbf{x}_i(t+1) = \mathbf{w}_i(t) - a_t [\mathbf{f}_i(\mathbf{z}_i(t+1)) + r_t \boldsymbol{\psi}_i(\mathbf{z}_i(t+1))]. \quad (33)$$

Hierbei ist a_t eine zeitabhängige Schrittweite, \mathbf{f}_i bzw. $\boldsymbol{\psi}_i$ bezeichnen die (Sub-)Gradienten der Funktion F_i bzw. Ψ_i . Der Wert der Optimierungsvariablen zum Zeitpunkt t am Knoten i wird durch die Variable $\mathbf{z}_i(t)$ beschrieben, die aus dem Verhältnis der Summenvariablen $\mathbf{w}_i(t)$ und der Gewichtsvariablen $y_i(t)$ berechnet wird. Die Aktualisierungsgleichung (33) für den Zustand $\mathbf{x}_i(t)$ beinhaltet die aktuellen Gradienten der Zielfunktion. Damit kann der Algorithmus als Mittelwertbildung interpretiert werden, der durch die Gradienten so beeinflusst wird, dass die resultierenden Zustände \mathbf{z}_i der Lösung des betrachteten Optimierungsproblems entspricht. Diese Methodik hat den Vorteil, dass pro Iteration nur ein Kommunikationsschritt notwendig ist. Außerdem hängen die Aktualisierungsgleichungen nur von den Ausgangsgraden $|\mathcal{N}_j^+|$ der Knoten im Kommunikationsgraphen ab, die sich ebenfalls durch eine verteilte Methode berechnen lassen [3].

Wie in [14] bewiesen wurde, erreichen die Zustände $\mathbf{z}_i(t)$ einen Konsensus für $t \rightarrow \infty$. Der Wert dieses Konsensus \mathbf{z}_c entspricht der Lösung \mathbf{z}^* des allgemeinen, verteilten und beschränkten Problems aus Gleichung (25), d. h., es gilt

$$\lim_{t \rightarrow \infty} \mathbf{z}_i(t) = \mathbf{z}_c = \mathbf{z}^* \text{ für alle } i \in \mathcal{V}, \quad (34)$$

wenn die folgenden Annahmen zutreffen:

Annahme 2. Die Funktion F aus Problem (25) ist inf-kompakt. D. h., $\{\mathbf{x} \in \mathbb{R}^d : F(\mathbf{x}) \leq A\}$ ist kompakt für alle $A \in \mathbb{R}$.

Annahme 3. Der Kommunikationsgraph \mathcal{G} ist stark verbunden.

Bemerkung: Laut [14] konvergiert der Algorithmus auch bei zeitvarianten Graphen $G(t)$ zuverlässig gegen das Optimum. In diesem Fall wird vorausgesetzt, dass die Vereinigung der Graphensequenz $\{G(t)\}$ alle B Zeitintervalle stark verbunden ist. In der vorliegenden Arbeit werden für die Untersuchungen des Algorithmus allerdings ausschließlich statische Graphen verwendet.

Annahme 4. Die Gradienten \mathbf{f}_i und ∇c_i sind auf \mathbb{R}^d für alle $i \in [n]$ gleichmäßig beschränkt und Lipschitz kontinuierlich.

Annahme 5.

$$\begin{aligned} a_t &\leq a_s \text{ für alle } t \geq s, & \sum_{t=0}^{\infty} a_t &= \infty, \\ r_t &\geq 1, & r_t &\rightarrow \infty, \\ \sum_{t=0}^{\infty} a_t^2 r_t^3 &< \infty, & r_{t+1} - r_t &= o(a_t), a_t \rightarrow 0. \end{aligned}$$

Angewendet auf das in Abschnitt 3 vorgestellte Energie-Management-Problem ergeben sich die lokalen Kostenfunktionen

$$F_i(\mathbf{z}) = \begin{cases} K_i(p_i), & i \in \mathcal{N}_G, \\ -U_i(p_i), & i \in \mathcal{N}_V, \end{cases} \quad (35)$$

mit $\mathbf{z} = [p_1, \dots, p_n]$. Die Straffunktionen $\Psi_i(\mathbf{z})$ ergeben sich der Gleichung (26) und den Nebenbedingungen aus (9) und (10) entsprechend.

Im Gegensatz zum Verfahren in Abschnitt 4 unterscheidet der in diesem Abschnitt beschriebene Algorithmus nicht zwischen globalen und lokalen Nebenbedingungen. Damit wird das gesamte Optimierungsproblem vollständig global auf dem Netzwerk gelöst. Durch die allgemeine Formulierung lässt sich eine große Anzahl an verteilten Optimierungsproblemen lösen, insofern sie die Annahmen (2) bis (5) erfüllen. Allerdings kann durch die generalisierte Anwendbarkeit nicht explizit auf Eigenschaften von einzelnen Optimierungsproblemen eingegangen werden, wie beispielsweise den hohen Anteil von lokalen Funktionen des in dieser Arbeit betrachteten Energie-Management-Problems.

6 Simulationen

Im Folgenden wird die Anwendung des in Abschnitt 5 vorgestellten Gradientenverfahrens mit Straffunktion auf das in Abschnitt 3 beschriebene verteilte Energie-Management-Problem untersucht. Dazu werden zwei Szenarien definiert, die sich in Anzahl und Konfiguration der Agenten voneinander unterscheiden.

Das erste Szenario umfasst vier Agenten mit zwei Generatoren und zwei Verbrauchern. Die Parametrierung entspricht den jeweils ersten zwei Werten für Generatoren und Verbrauchern aus der Tabelle 1. Die Agenten sind über den in Abbildung 1 dargestellten Kommunikationsgraphen untereinander verbunden.

Das zweite Szenario umfasst acht Agenten, von denen vier vom Typ Generator und vier vom Typ Verbraucher sind. Die Parametrierung entspricht der Tabelle 1. Für dieses Szenario werden drei Kommunikationsgraphen untersucht, die sich in der Kantenanzahl und damit in der Anzahl der Kommunikationskanäle zwischen den Agenten voneinander unterscheiden. Die Graphen sind in den Abbildungen 2 bis 4 dargestellt.

Für beide Szenarien wird das Problem (6) zunächst zentral ohne Straffunktionen mit einem Standard-Solver

Tab. 1: Parametrierung der Agenten in den untersuchten Szenarien. Szenario 1: IDs 1-2, Szenario 2: IDs 1-4.

Parameter der Generatoren

ID, i	a	b	c	l	\underline{p}	\bar{P}
1	0,0024	5,56	30	0,00021	60	339,69
2	0,0056	4,32	25	0,00031	25	479,10
3	0,0072	6,60	25	0,00011	28	290,40
4	0,0047	3,14	16	0,00022	40	306,34

Parameter der Verbraucher

ID, j	ω	K	α	\underline{p}	\bar{P}
1	18,43	0,5	0,0545	50	100,34
2	13,17	0,5	0,0877	100	159,13
3	15,46	0,5	0,0547	40	80,56
4	10,03	0,5	0,1041	30	123,98

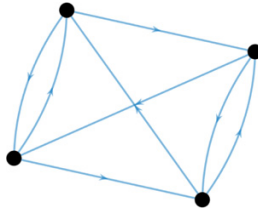


Abb. 1: Graph mit vier Knoten und acht Kanten.

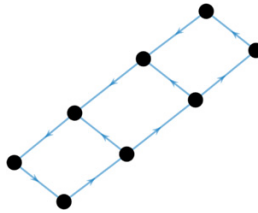


Abb. 2: Netzgraph mit acht Knoten und zehn Kanten.

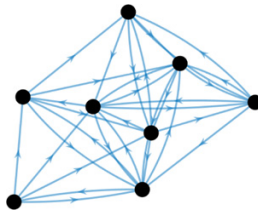


Abb. 3: Zufälliger Graph mit 8 Knoten und 32 Kanten.

gelöst, um das globale Optimum p_i^* zu berechnen. Auf dieses Resultat wird der relative Fehler

$$\delta_i = \frac{|p_i - p_i^*|}{p_i^*}, i \in \mathcal{V} \quad (36)$$

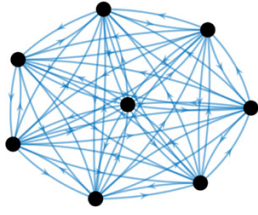


Abb. 4: Vollständiger Graph mit 8 Knoten und 56 Kanten.

der verteilten Lösung p_i bezogen. Zur Berechnung des Fehlers im verteilten Fall werden bei den folgenden Betrachtungen nur die Ergebnisse eines Agenten ausgewertet, d. h., es wird davon ausgegangen, dass alle Agenten sich annähernd miteinander im Konsensus befinden.

In der folgenden Analyse wird zunächst ein zentrales Gradientenverfahren mit Straffunktionsansatz zur Lösung von Szenario 1 betrachtet. Die Ergebnisse hieraus werden anschließend mit den Resultaten für das erste Szenario aus dem verteilten Gradientenverfahren mit Straffunktionen, vorgestellt in Abschnitt 5, verglichen. Schwerpunkt der Betrachtung liegt hierbei auf der Festlegung der Folgen

$$a_t = \frac{3}{t^{k_a}}, r_t = t^{k_r}. \quad (37)$$

durch die Parameter k_a und k_r , wobei vor allem die Konvergenzgeschwindigkeit und Stabilität des Verfahrens von Bedeutung sind. Anschließend wird das Energie-Management-Problem von Szenario 2 mit dem verteilten Verfahren gelöst und untersucht, wie sich die erhöhte Agentenanzahl und die unterschiedlichen Vernetzungsgrade des Kommunikationsgraphen auf die Konvergenzgeschwindigkeit auswirken.

6.1 Zentrale Lösung, Szenario 1: Gradientenverfahren mit Straffunktionen und Schrittweitensteuerung

Um einen Vergleichswert für die Konvergenzgeschwindigkeit des verteilten Gradientenverfahrens mit Straffunktionen zu erhalten, wird zunächst eine zentrale Variante dieser Optimierungsmethode auf das Energie-Management-Problem aus (7) angewendet. Damit wird davon ausgegangen, dass alle Instanzen im Netzwerk unmittelbar Zugriff auf alle Informationen über Zustände, Kostenfunktionen und Nebenbedingungen haben.

Um eine möglichst optimale, d. h. geringe, Iterationsanzahl des Algorithmus zu erhalten, wird die Schrittweite a_t des Gradientenverfahrens über die Minimierungsregel

[16] festgelegt. Hierbei wird in jedem Iterationsschritt des Algorithmus die Schrittweite über

$$a_t^{\text{opt}} = \operatorname{argmin}_{s \geq 0} F_t(\mathbf{z} - s \nabla f_t(\mathbf{z})) \quad (38)$$

bestimmt. Um die optimale Folge r_t des Strafparameters zu identifizieren, wird der relative Fehler nach 5000 Iterationen für unterschiedliche Parametrierungen von k_r aus Gleichung (37) berechnet und der Parameter mit dem geringsten Fehler ausgewählt. Durch diese Methodik ergibt sich für das Szenario 1 mit vier Agenten die Folge $r_t = t^{0,18}$ im zentralen Fall. Der Verlauf dieser Folge über die Iterationen der Optimierung ist in Abbildung 7 dargestellt.

In Abbildung 5 ist der Verlauf der relativen Fehler δ_i zwischen den optimalen Werten p_i^* und den Werten $p_i(t)$ des Algorithmus zum Zeitpunkt t dargestellt. Das Verfahren wird abgebrochen, nachdem alle relativen Fehler einen Wert von $\delta_i = 0,015$ unterschreiten.

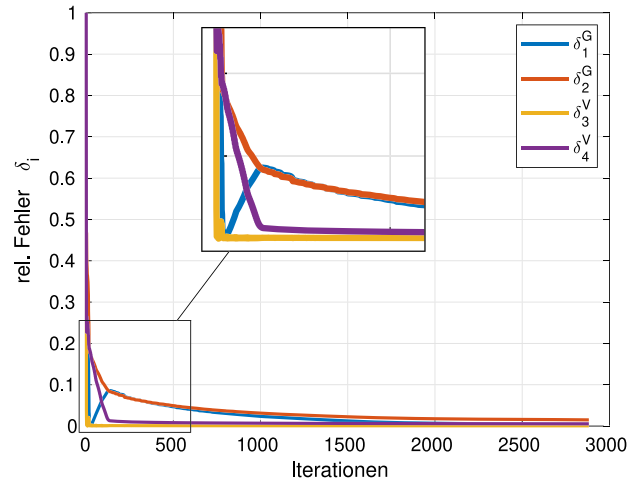


Abb. 5: Verlauf der relativen Fehler $\delta_i, i \in \mathcal{V}$ für zentrales Gradientenverfahren mit Straffunktionen und Schrittweitenadaption. Konvergenz auf relativen Fehler $\delta_i < 0,015$ für alle $i \in \mathcal{V}$ nach 2881 Iterationen.

In der Graphik kann erkannt werden, dass bereits nach 107 Iterationen ein relativer Fehler von 0,1 unterschritten wird. Besonders die Variablen der Verbraucher nähern sich dem Optimum sehr schnell und genau an, so dass der Fehler bereits nach 265 Iterationen kleiner als 0,01 ist. Die Generatoren hingegen konvergieren nach etwa 100 Iterationen nur noch sehr langsam zum Optimum, was die hohe Iterationsanzahl des Verfahrens bedingt.

Zur Erläuterung dieses Verhaltens ist in Abbildung 6 die Auswertung der globalen Nebenbedingung aus Gleichung (6a) über die Zeit abgebildet. Es kann erkannt werden, dass nach etwa 100 Iterationen die Annäherungsgeschwindigkeit an den Wert Null deutlich schwächer wird.

Offensichtlich sind durch die Parametrierung des Problems hauptsächlich die Generatoren für die Erfüllung dieser Nebenbedingung verantwortlich, die mit der Optimierung der eigentlichen Zielfunktion, der Minimierung der Kosten, konkurriert. In Abbildung 7 kann erkannt werden, dass sich der Faktor r_t in diesem Bereich nur noch langsam ändert. Dies lässt vermuten, dass sich die Konvergenzgeschwindigkeit durch eine mögliche Adaption der Strafparameterfolge weiter verbessern lässt.

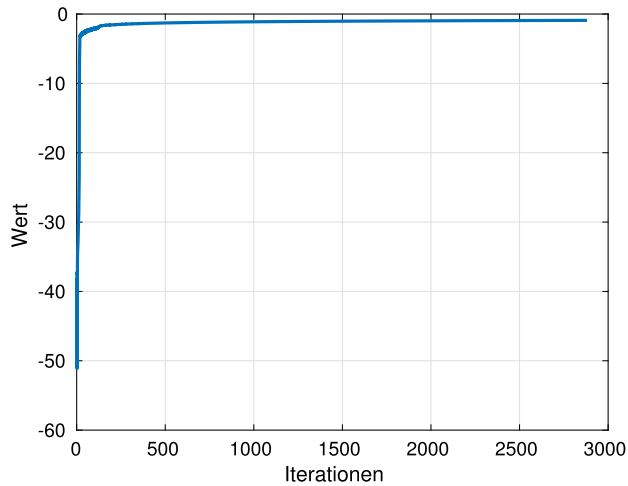


Abb. 6: Verlauf der globalen Nebenbedingung aus Gleichung (6a) für zentrales Gradientenverfahren mit Schrittweitensteuerung über die Zeit.

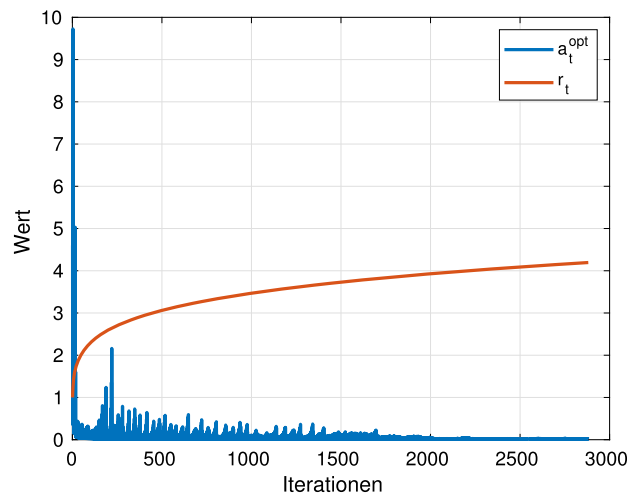


Abb. 7: Adaptierte Schrittweite a_t^{opt} und Straffolge r_t mit $k_r = 0,18$ für zentrales Gradientenverfahren.

In Abbildung 8 ist der Gradientenverlauf ∇F_t über die Iterationen dargestellt. Es kann erkannt werden, dass gerade zu Beginn der Gradient stark schwankt und über den

kompletten Verlauf keine monotone Abnahme zu erkennen ist. Dieses Phänomen entsteht durch das Aktivieren und Deaktivieren der Straffunktionen, was wiederum aus der Verletzung einer oder mehrerer Nebenbedingungen resultiert. Dadurch kann der Gradient zwischen zwei aufeinander folgenden Iterationen trotz der streng konvexen Kostenfunktion und der Schrittweitensteuerung stark ansteigen oder abfallen. Wird vergleichend die Abbildung 7 betrachtet, kann erkannt werden, dass die adaptierte Schrittweite in Bereichen hoher Schwankungen im Gradienten (Iterationen 1 bis 150) relativ niedrig ist. Damit fängt die Schrittweite plötzliche Änderungen im Gradienten ab und vermindert dadurch die Auswirkungen auf den Zustands- bzw. den Fehlerverlauf aus Abbildung 5.

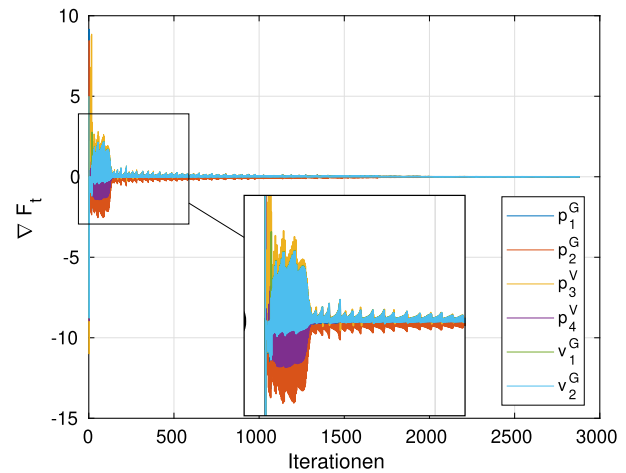


Abb. 8: Verlauf des Gradienten ∇F_t für zentrales Verfahren mit Schrittweitensteuerung.

6.2 Verteilte Lösung, Szenario 1: Wahl der Schrittweite und des Strafparameters

In diesem Unterabschnitt wird die verteilte Lösung des Problems mit den Ergebnissen des zentralen Gradientenabstiegs aus dem vorherigen Abschnitt für das Szenario 1 verglichen. Im Vordergrund steht hier eine möglichst optimale Wahl der Schrittweite a_t des Gradientenverfahrens und die Wahl der Folge r_t des Strafparameters. Durch den Vergleich mit dem zentralen Verfahren sollen Rückschlüsse auf die korrekte Wahl der Folgen a_t und r_t im verteilten Szenario möglich werden. In [14] wird eine Empfehlung für die Wahl dieser Folgen angegeben, unter denen das Verfahren garantiert im Unendlichen auf das Optimum konvergiert. In den Versuchen zeigt sich allerdings, dass die angegebene Schrittweite praktisch sehr hohe Konvergenzzeiten nach sich zieht, weswegen die annähernd optimale

Schrittweite und Folge des Strafparameters im Folgenden experimentell bestimmt wird.

Grundsätzlich werden beide Folgen durch die Vorschrift in Gleichung (37) bestimmt, wobei die Parameter k_a und k_r die jeweilige Folge festlegen. Zwei Konvergenzeigenschaften werden von deren Wahl beeinflusst. Im Wesentlichen ist die Iterationsanzahl bis zur Reduzierung des relativen Fehlers auf einen festgelegten Wert von dem Verhältnis von a_t und r_t abhängig. Gleichzeitig muss jedoch die Schwingung des Gradienten berücksichtigt werden, die sich auch auf die Zustände überträgt, sodass sich bei bestimmten Parameterkonfigurationen dauerhafte Schwingungen um den optimalen Wert einstellen.

Um einen möglichst optimalen Bereich, in dem das Verhältnis von k_a und k_r liegen darf, festzulegen, werden in den Versuchen verschiedene Parameterkonfigurationen untersucht. In Abbildung 9 ist der über alle Zustände summierte relative Fehler über k_r für einen festen Parameter $k_a = 0,35$ dargestellt. Grün markiert ist der Bereich, in dem der Parameter liegen sollte, damit zum einen eine schnelle Konvergenz der Optimierung gewährleistet ist und zum anderen keine oder nur geringe Schwingungen auftreten. Es ist festzustellen, dass nur ein schmaler Bereich für die Auswahl von k_r in Frage kommt.

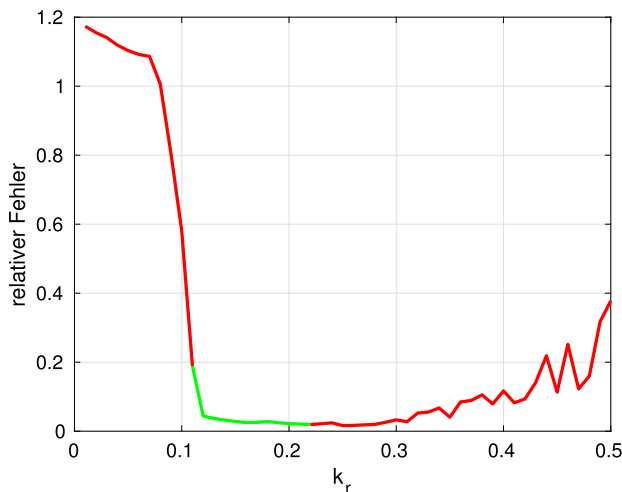


Abb. 9: Verlauf des aufsummierten relativen Fehlers $\sum_i \delta_i$ über die Wahl des Parameters k_r bei fester Wahl von $k_a = 0,16$.

In Abbildung 10 ist der Verlauf des relativen Fehlers über die Iterationen mit der Wahl $k_a = 0,35$ und $k_r = 0,16$ für die Faktoren der Folgen dargestellt. Durch die feste Vorgabe der Folge a_t kann für das verteilte Szenario 1 nicht flexibel auf eine hohe Variabilität im Gradienten reagiert werden. Hieraus ergibt sich im Vergleich zu dem zentralen Verfahren mit Schrittweitensteuerung, vgl. Abbildung 5, im

Anfangsbereich bis 700 Iterationen eine längere Dauer bis zur Absenkung des Fehlers unter 0,1. Nach 700 Iterationen zeigt sich ein ähnlicher Verlauf wie im zentralen Fall. Werden die Unterschiede im Fehlerverlauf mit den Unterschieden in der Schrittweite verglichen, so fällt auf, dass die größten Unterschiede des Fehlers genau mit dem Bereich der größten Unterschiede der Schrittweiten zusammenfallen. Dies lässt den Schluss zu, dass durch die harte Vorgabe der Schrittweitenfolge im verteilten Fall nicht optimal, d. h., nicht flexibel genug, auf die Aktivität der Straffunktion reagiert werden kann. Daraus lässt sich die Vermutung ableiten, dass durch eine verteilte Schrittweitenadaption eine Verbesserung der Konvergenzrate erzielt werden könnte. Außer acht gelassen wird bei diesem Schluss die Konsensus-Dynamik und damit die Unterschiede in den jeweiligen lokalen Gradientenabstiegen. Da der Kommunikationsgraph aus Abbildung 1 sehr gut vernetzt ist und zusätzlich das Szenario mit nur vier Agenten überschaubar und somit der Verteilungsgrad des Problems nicht allzu hoch ist, wird davon ausgegangen, dass der Schluss Bestand hat. Da für eine optimale Schrittweitenadaption die komplette Zielfunktionsauswertung notwendig ist, kann die Adaption nicht direkt auf den verteilten Fall übertragen werden.

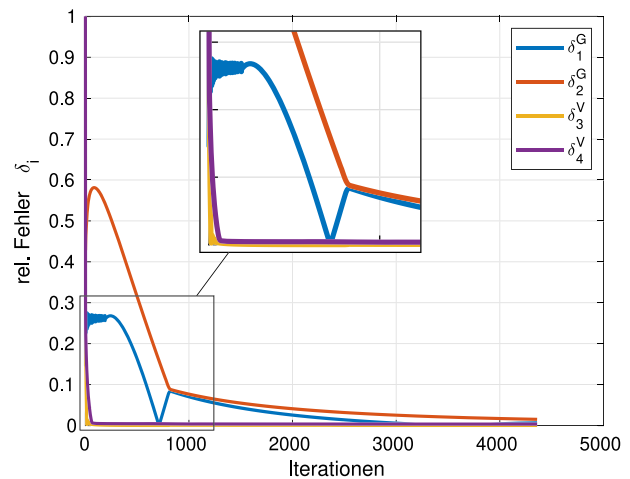


Abb. 10: Verlauf der relativen Fehler $\delta_i, i \in \mathcal{V}$ für verteiltes Gradientenverfahren mit Straffunktionen und festen Werten $k_a = 0,35$, $k_r = 0,16$. Konvergenz auf relativen Fehler $\delta_i < 0,015$ für alle $i \in \mathcal{V}$ nach 4360 Iterationen.

Insgesamt kann aus diesem Unterabschnitt festgehalten werden, dass das Konfigurieren der Folgen a_t und r_t bereits bei kleinen Szenarien aufwendig ist, da keine allgemein gültige Regel existiert. Verantwortlich sind dafür die Abhängigkeiten von Kommunikationsgraph, Konsensus-

Dynamik und Agentenanzahl. Aus diesem Grund empfiehlt sich eine verteilte Adaption der Schrittweite, die Gegenstand zukünftiger Arbeiten sein kann.

6.3 Verteilte Lösung, Szenario 2: Einfluss des Graphen und höherer Knotenanzahlen auf Konvergenzzeit

In diesem Unterabschnitt wird untersucht, welchen Einfluss die Größe des verteilten Optimierungsproblems und die Verbindungsdichte im Kommunikationsgraphen auf die Konvergenz des Verfahrens hat. Dazu wird das eingangs beschriebene zweite Szenario betrachtet und die festen Folgen $\alpha_t = 3/t^{0,35}$ und $r_t = t^{0,19}$ für alle Simulationen gewählt.

Zunächst wird der Einfluss des Verbindungsgrades des Kommunikationsgraphen auf die Konvergenzzeit betrachtet. Verglichen werden dazu ein Gittergraph mit zehn Verbindungen, ein zufälliger Graph mit 32 Verbindungen und ein vollständiger Graph mit 56 Verbindungen, siehe Abbildungen 2 bis 4. Auf diesen Graphen wird das verteilte Gradientenverfahren für Szenario 2 angewendet, wobei die Iterationen abgebrochen werden, nachdem alle relativen Fehler kleiner als 0,02 sind. In der Tabelle 2 sind die Iterationsanzahlen, die sich für die jeweiligen Graphen ergeben, aufgeführt. Es kann eine deutliche Iterationsreduktion zwischen dem Graphen mit zehn Verbindungen und dem Graphen mit 32 Verbindungen erkannt werden. Der Sprung zum vollständigen Graphen ist mit einer Differenz von nur 66 Iterationen allerdings unwesentlich. Folgerung lässt sich hieraus, dass der Verbindungsgrad zwar bei sehr wenigen Verbindungen eine große Rolle spielt, da die Durchsetzung des Netzwerkes mit Informationen wesentlich länger dauert. Gleichzeitig steht die Konsensgeschwindigkeit bei einem guten Verbindungsgrad nicht mehr im Vordergrund. Bereits in [10] wird auf die hohe Konsensus-Geschwindigkeit auf sogenannten Small-World-Graphen hingewiesen, zu denen auch der Graph mit 32 Verbindungen gehört. Die Folgerung bestätigt sich auch bei Betrachtung des vollständig verbundenen Graphen. Hier kann jeder Agent mit jedem anderen Agent Informationen austauschen, was genau genommen nicht mehr einem verteilten Problem entspricht. Trotzdem wird hier eine hohe Anzahl von Iterationsschritten bis zu einer annähernden Konvergenz benötigt, die sich kaum von dem Graphen mit 32 Verbindungen unterscheidet.

Damit kann festgestellt werden, dass die Methodik des verteilten Gradientenabstiegs mit Straffunktionen selbst den Engpass für die Iterationsanzahl darstellt bzw. die Problematik der optimalen Folgenwahl, die im vorherigen Ab-

Tab. 2: Anzahl der Iterationen bis zum unterschreiten eines relativen Fehlers von 0,02 für Graphen mit steigendem Verbindungsgrad.

	$ \mathcal{E} = 10$	$ \mathcal{E} = 32$	$ \mathcal{E} = 56$
Iter.	14953	12011	11945

schnitt bereits diskutiert wurde, ausschlaggebend für die Konvergenzzeit ist.

In Abbildung 11 ist der Verlauf des relativen Fehlers für den Graphen mit 32 Verbindungen über die Iterationen dargestellt. Ähnlich wie im Fall des Szenarios 1 aus Abbildung 10 bedingen hauptsächlich zwei Agenten die lange Konvergenzzeit von 12011 Iterationen. Diese ist bei einer Verdopplung der Agentenanzahl von Szenario 1 zu Szenario 2 fast dreimal so hoch wie die für Szenario 1. Damit lässt sich festhalten, dass das Verfahren trotz Verwendung eines Konsensus-Protokolls, das an sich eine gute Skalierungseigenschaft aufweist, selbst nicht optimal skaliert. Es wird vermutet, dass dies aus der Erhöhung der Straffunktionsanzahl resultiert, deren wiederholtes Aktivieren und Deaktivieren ein gleichmäßiges Konvergieren des Verfahrens verhindert, was durch eine konstante Wahl von α_t auch nicht von der Schrittweite abgefangen werden kann.

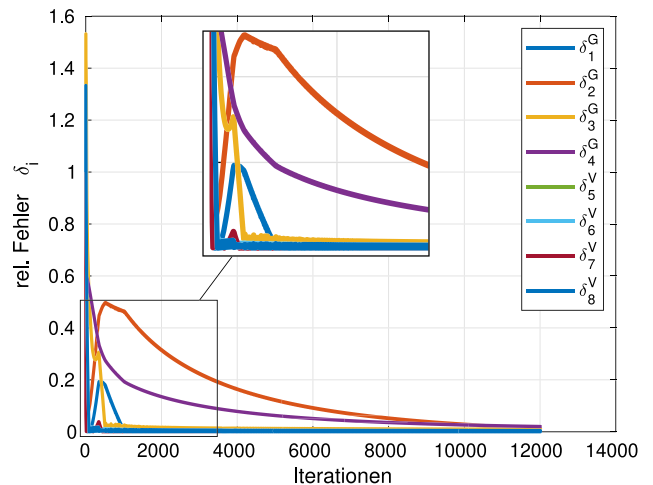


Abb. 11: Verlauf der relativen Fehler $\delta_i, i \in \mathcal{V}$ für zentrales Gradientenverfahren mit Straffunktionen und Schrittweitensteuerung. Konvergenz auf relativen Fehler $\delta_i < 0,02$ für alle $i \in \mathcal{V}$ nach 12011 Iterationen.

Zusammengefasst kann festgehalten werden, dass der Verbindungsgrad ab einer bestimmten Höhe sich in diesem Szenario nicht mehr ausschlaggebend restriktiv auf die Konvergenzgeschwindigkeit auswirkt. Dies steht in Zusammenhang mit der schlechten Skalierung der Itera-

tionsanzahl bei Vergrößerung des Umfangs des Energie-Management-Problems.

7 Fazit

Das untersuchte verteilte Gradientenverfahren mit Beschränkungen durch Straffunktionen zeichnet sich durch seine hohe Generalität aus. Dadurch ist es möglich, eine große Menge an konvexen und verteilten Optimierungsproblemen näherungsweise zu lösen. Durch die Eingliederung des Gradientenabstiegs in das Push-Sum-Protokoll konvergiert der Algorithmus auch bei gerichteten und zeitvarianten Kommunikationsstrukturen zuverlässig gegen das Optimum. Der Nachteil des Verfahrens besteht in seiner vergleichsweise langsamen Konvergenz, die sich auch bei einer zentralen Ausführung beobachten lässt. Damit stellt nicht die verteilte Ausführung des Verfahrens den Konvergenzengpass dar, sondern die Beschränkungen des Problems durch Straffunktionen. Das verteilte Szenario verhindert jedoch zunächst eine Verwendung von beschleunigenden Techniken, wie einer Schrittweitenadaptation. Die verteilte Ausführung einer solchen kann Gegenstand von zukünftigen Arbeiten sein.

Daneben kann eine geschickte Ausnutzung von ausschließlich lokalen Beschränkungen in verteilten Energie-Management-Problemen dazu verwendet werden, den Kommunikationsaufwand zwischen den Agenten zu verringern, wie in Abschnitt 4 diskutiert. Hierzu muss die bestehende Methode aus [18] so erweitert werden, dass sie in jedem Fall auf das Optimum des Energie-Management-Problems konvergiert.

Faktoren, die bisher in den wenigsten verteilten Problemen berücksichtigt wurden, sind unsichere Datenübertragungen und verrauschte Messungen. Ist das Ziel, das vorgestellte Energie-Management-Problem der Praxis näher zu bringen, müssen solche stochastischen Beeinflussungen berücksichtigt werden. Dazu können beispielsweise die Ergebnisse für unsichere und verrauschte Konsensusprotokolle aus [17] aufgegriffen und auf verteilte Optimierungsprobleme bezogen werden.

Finanzierung: Diese Arbeit wurde gefördert durch die Deutsche Forschungsgemeinschaft (DFG) – Schwerpunktprogramm 1984, Projektnummer 16.

Literatur

1. M. Ahat, S. B. Amor, M. Bui, A. Bui, G. Guérard and C. Petermann. Smart Grid and Optimization. *American Journal of Operations Research*, 03(01):196–206, 2013.
2. T. Charalambous and C. N. Hadjicostis. Average consensus in the presence of dynamically changing directed topologies and time delays. In *53rd IEEE Conference on Decision and Control*, pages 709–714, Los Angeles, CA, USA, Dec. 2014. IEEE.
3. T. Charalambous, M. G. Rabbat, M. Johansson and C. N. Hadjicostis. Distributed Finite-Time Computation of Digraph Parameters: Left-Eigenvector, Out-Degree and Spectrum. *IEEE Transactions on Control of Network Systems*, 3(2):137–148, June 2016.
4. D. Jakovetic, D. Bajovic, N. Krejic and N. Krklec Jerinkic. Distributed Gradient Methods with Variable Number of Working Nodes. *IEEE Transactions on Signal Processing*, 64(15):4080–4095, Aug. 2016.
5. D. Jakovetic, J. Xavier and J. M. F. Moura. Fast Distributed Gradient Methods. *IEEE Transactions on Automatic Control*, 59(5):1131–1146, May 2014.
6. A. Kargarian, J. Mohammadi, J. Guo, S. Chakrabarti, M. Barati, G. Hug, S. Kar and R. Baldick. Toward Distributed/Decentralized DC Optimal Power Flow Implementation in Future Electric Power Systems. *IEEE Transactions on Smart Grid*, 9(4):2574–2594, July 2018.
7. D. Kempe, A. Dobra and J. Gehrke. Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings*, pages 482–491, Cambridge, MA, USA, 2003. IEEE Computer. Soc.
8. A. Nedić and A. Olshevsky. Distributed Optimization Over Time-Varying Directed Graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, Mar. 2015.
9. A. Nedić, A. Ozdaglar and P. Parrilo. Constrained Consensus and Optimization in Multi-Agent Networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, Apr. 2010.
10. R. Olfati-Saber. Ultrafast consensus in small-world networks. In *Proceedings of the 2005, American Control Conference, 2005*, pages 2371–2378, Portland, OR, USA, 2005. IEEE.
11. B. T. Polyak. *Introduction to optimization*. Translations series in mathematics and engineering, 1984.
12. S. Pu, W. Shi, J. Xu and A. Nedic. Push-Pull Gradient Methods for Distributed Optimization in Networks. *arXiv:1810.06653 [cs, math]*, Oct. 2018.
13. P. Siano. Demand response and smart grids-A survey. *Renewable and Sustainable Energy Reviews*, 30:461–478, Feb. 2014.
14. T. Tatarenko, J. Zimmermann, V. Willert and J. Adamy. Penalized Push-Sum Algorithm for Constrained Distributed Optimization with Application to Energy Management in Smart Grid. In: *2019 IEEE 58th Conference on Decision and Control (CDC)*, Nice, France, December 2019.
15. K. I. Tsianos, S. Lawlor and M. G. Rabbat. Push-Sum Distributed Dual Averaging for convex optimization. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 5453–5458, Maui, HI, Dec. 2012. IEEE.
16. M. Ulbrich und S. Ulbrich. *Nichtlineare Optimierung*. Mathematik kompakt. Birkhäuser, Basel, 2012. OCLC: 799006886.
17. V. Willert, D. Haumann and S. Gering. Decentralized Bayesian consensus over networks. In *2014 European Control Conference (ECC)*, pages 1600–1606, Strasbourg, France, June 2014. IEEE.
18. C. Zhao, J. He, P. Cheng and J. Chen. Consensus-Based Energy Management in Smart Grid With Transmission Losses and Directed Communication. *IEEE Transactions on Smart Grid*, 8(5):2049–2061, Sept. 2017.

Autoreninformationen



Jan Zimmermann

Technische Universität Darmstadt,
Fachbereich Elektrotechnik und
Informationstechnik, Fachgebiet
Regelungsmethoden und Robotik, D-64283
Darmstadt, Deutschland
jan.zimmermann@rmr.tu-darmstadt.de

M. Sc. Jan Zimmermann ist wissenschaftlicher Mitarbeiter am Fachgebiet Regelungsmethoden und Robotik unter der Leitung von Prof. Dr.-Ing. J. Adamy am Institut für Automatisierungstechnik und Mechatronik im Fachbereich Elektrotechnik und Informationstechnik der Technischen Universität Darmstadt. Hauptarbeitsgebiete: Verteilte Optimierung, Spieltheorie.



Tatiana Tatarenko

Technische Universität Darmstadt,
Fachbereich Elektrotechnik und
Informationstechnik, Fachgebiet
Regelungsmethoden und Robotik, D-64283
Darmstadt, Deutschland
tatiana.tatarenko@rmr.tu-darmstadt.de

Dr. rer. nat. Tatiana Tatarenko ist Gruppenleiterin der Forschergruppe *Verteilte Optimierung und Spieltheorie* des Fachgebietes Regelungsmethoden und Robotik unter der Leitung von Prof. Dr.-Ing. J. Adamy am Institut für Automatisierungstechnik und Mechatronik im Fachbereich Elektrotechnik und Informationstechnik der Technischen Universität Darmstadt. Hauptarbeitsgebiete: Angewandete Mathematik, verteilte Optimierung, Spieltheorie.



Volker Willert

Technische Universität Darmstadt,
Fachbereich Elektrotechnik und
Informationstechnik, Fachgebiet
Regelungsmethoden und Robotik, D-64283
Darmstadt, Deutschland
vwillert@rmr.tu-darmstadt.de

Dr.-Ing. Volker Willert ist Gruppenleiter der Forschergruppe *Autonome Systeme und Mobile Robotik* des Fachgebietes Regelungsmethoden und Robotik unter der Leitung von Prof. Dr.-Ing. J. Adamy am Institut für Automatisierungstechnik und Mechatronik im Fachbereich Elektrotechnik und Informationstechnik der Technischen Universität Darmstadt. Hauptarbeitsgebiete: Inferenzmethoden in der Regelungstechnik, maschinelles Sehen, Mustererkennung, Multi-Agenten-Systeme.



Jürgen Adamy

Technische Universität Darmstadt,
Fachbereich Elektrotechnik und
Informationstechnik, Fachgebiet
Regelungsmethoden und Robotik, D-64283
Darmstadt, Deutschland
adamy@rmr.tu-darmstadt.de

Prof. Dr.-Ing. Jürgen Adamy ist Professor des Fachgebietes Regelungsmethoden und Robotik am Institut für Automatisierungstechnik und Mechatronik im Fachbereich Elektrotechnik und Informationstechnik der Technischen Universität Darmstadt. Hauptarbeitsgebiete: Nichtlineare Regelungstechnik, Computational Intelligence, Mobile Robotik.